

AD-A174 860

AN EFFICIENT ALGORITHM TO CLUSTER ORDER PICKING ITEMS
IN A WIDE AISLE. (U) GEORGIA INST OF TECH ATLANTA
PRODUCTION AND DISTRIBUTION RESE.

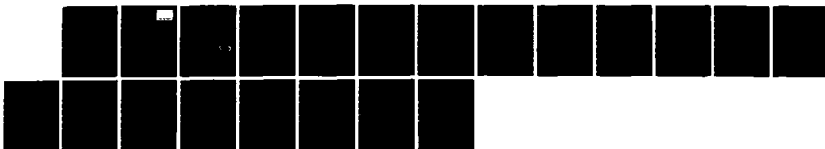
1/1

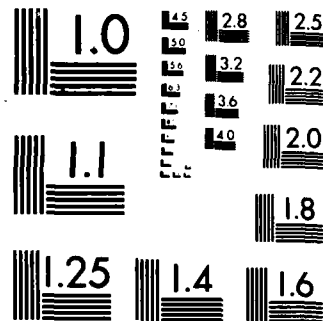
UNCLASSIFIED

M GOETSCHALCKX ET AL. OCT 86 PDRC-86-12

F/G 15/5

ML





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(12)

AD-A174 060

DTIC
ELECTE
NOV 12 1986

D

AF

Georgia Institute
of
Technology



AN EFFICIENT ALGORITHM TO CLUSTER ORDER
PICKING ITEMS IN A WIDE AISLE

Marc Goetschalekx
H. Donald Ratliff

PDRG 86-12

**PRODUCTION and
DISTRIBUTION RESEARCH
CENTER**

DTIC FILE COPY

DISTRIBUTION STATEMENT A

Approved for public release
Distribution Unlimited

OFFICE OF INDUSTRIAL
AND SYSTEM
ENGINEERING
GEORGIA INSTITUTE OF TECHNOLOGY
3140 DECATUR AVENUE, N.W.
ATLANTA, GEORGIA 30332

86 11 12 079

AN EFFICIENT ALGORITHM TO CLUSTER ORDER
PICKING ITEMS IN A WIDE AISLE

Marc Goetschalckx
H. Donald Ratliff

PDRC 86-12

DTIC
ELECTE
NOV 12 1986
S D D

School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332

This work was supported by the Office of Naval Research under Contract No.
N00014-86-K-0173. Reproduction is permitted in whole or in part for any purpose of
the U. S. Government.

DISTRIBUTION STATEMENT A

Approved for public release
Distribution Unlimited

AN EFFICIENT ALGORITHM TO CLUSTER ORDER PICKING ITEMS IN A WIDE AISLE.

Marc Goetschalckx
H. Donald Ratliff
Georgia Institute of Technology
Atlanta, Georgia 30332-0205

0. ABSTRACT

In many warehousing systems cases of items have to be retrieved manually from both sides of a wide aisle and deposited on a vehicle which travels on the center line of the aisle. The picker will stop the vehicle, pick and load cases of items onto the vehicle and then drive to the next stop. There is a tradeoff between the time to stop and start the vehicle and the increased walking distance if fewer vehicle stops are made.

An algorithm is presented to optimally determine the number and location of stops and to specify the items to be picked at each stop. Comparison with heuristic clustering methods indicates savings of 10 % to 90 %. The algorithm can be implemented in real time on a microcomputer.

KEYWORDS : order picking, set covering, material handling.



| | |
|--------------------|--|
| Accession For | |
| NTIS CRA&I | <input checked="checked" type="checkbox"/> |
| DTIC TAB | <input type="checkbox"/> |
| Unannounced | <input type="checkbox"/> |
| Justification | |
| By _____ | |
| Distribution / | |
| Availability Codes | |
| Dist | Avail and/or Special |
| A-1 | |

1. INTRODUCTION

One of the most common order picking systems is case picking, where products are stored on pallets on both sides of a wide aisle. The aisles are usually 12 or more feet wide to allow the pallets to be stored by fork lift truck. Frequently, the picking vehicle is a tug pulling two pallets on dollies or a pallet jack with a capacity of two pallets. Such vehicles cannot make sharp turns, hence the picking is done by driving down the center of the aisle with periodic stops for picking. The vehicle is stopped on the center line of the aisle. All the cases to be picked at the stop are manually retrieved one-at-a-time and placed on the vehicle. The vehicle is then driven to its next stop. Prominent examples of such systems occur in grocery distribution and parts distribution systems. For other order picking systems in wide aisles, see Goetschalckx and Ratliff (1985).

The objective is to maximize order throughput or equivalently minimizing the overall picking time. The overall picking time consists of (a) the walking time of the driver, (b) the time to start and stop the vehicle one or more times, (c) the vehicle driving time (d) the time to pick the items, and (e) the time to stack the items on the vehicle. It will be assumed here that (c), (d) and (e) are independent of the number of stops and their location. The problem is then to determine how many stops should be made, where the vehicle should be stopped and which picks should be associated with each stop, in order to minimize the sum of (a) the time spent walking between vehicle and the pick and (b) the time spent starting and stopping the vehicle. An illustration of the picking problem is given in Fig 1.

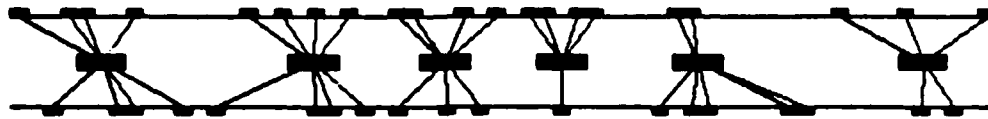


Figure 1. Wide Aisle Picking Illustration.

In section 2 a procedure is developed to determine for each order the optimal stops and associated picks at each stop. In Section 3, two procedures are developed to determine the stops if the same stops are to be used for all orders. All three procedures are implemented on a microcomputer. They are compared experimentally in Section 4 with respect to computation time and travel time.

2. OPTIMAL LOCATION OF VEHICLE STOPS

It is assumed that the vehicle travels on the center line of the aisle and is stopped on the center line. Two different assumptions are considered with respect to travel between the picking vehicle and the pick. First, it is assumed that the picker travels in straight line from the vehicle to the item on the pallet and back. This straight line travel is also called Euclidean travel. Next, it is assumed that the picker travels first to the side of the aisle and then along the side of the aisle. This travel pattern is also called rectilinear travel. The actual travel is a combination of both and is more like rectilinear for travel from the vehicle to the first pick at a stop and more like Euclidean for the remaining picks at a stop. An illustration of Euclidean (dashed line) and rectilinear (solid line) travel to a pick is shown in Figure 2. It will be shown that optimal stops do not depend significantly on which of the two travel assumptions is selected.

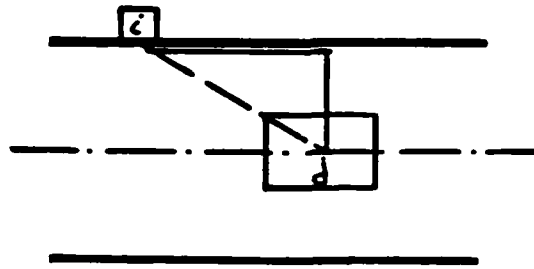


Figure 2. Euclidean and Rectilinear Travel to a Pick.

Let a_j denote the distance along the aisle center line from the beginning of the aisle to case i . Let z_j denote the distance along the aisle center line from the beginning of the aisle to vehicle stop j . Let w denote the aisle width. Let d_{ij} denote the one-way distance (either Euclidean or rectilinear) between item i and vehicle stop j . Let f denote the time it takes to stop and start the vehicle, i.e. it is the incremental time required for each vehicle stop. Let N

be the number of items in the order. Let v be the walking speed of the picker between the vehicle and the picks.

The following properties allow an efficient optimum procedure to be established for determining the stops, assuming either Euclidean or rectilinear travel between the vehicle and the pick.

Property 1.

Given a set of vehicle stops, any optimal procedure will always pick each item from its closest vehicle stop.

Proof The proof is by contradiction. Assume an item is not assigned to its closest stop by an optimal procedure. Assigning the item to its closest vehicle stop would decrease the retrieval time for that item and not increase any other travel times. Hence, the current assignment cannot be optimal. Q.E.D.

Property 2.

All the items, picked from a single vehicle stop by an optimal algorithm, are located consecutively along both sides of the aisle, without interruption by items picked from different vehicle stops.

Proof The proof follows immediately from Property 1.

Hence, an optimal algorithm must have consecutive blocks of items associated with each stop. The total time associated with a single vehicle stop, denoted by c_j , is the total round trip travel time from all the picks associated with this stop to the optimal vehicle location plus the fixed vehicle start/stop time. Let P_j denote the set of picks assigned to stop j , then

$$c_j = f + 2 \sum_{i \in P_j} d_{ij} / v.$$

Property 3.

Given the items to be picked from a stop, the optimum location of the vehicle stop assuming Euclidean travel distance is given by the following recursive formula.

$$z_j = \frac{\sum_{i \in P_j} a_i / d_{ij}}{\sum_{i \in P_j} 1 / d_{ij}},$$

where

$$d_{ij} = \sqrt{(a_i - z_j)^2 + (w/2)^2}.$$

Proof This is the equivalent of a single facility Euclidean location problem, Francis and White(1974), Chapter 4. Q.E.D.

Observe that all d_{ij} are always positive, since the vehicle is located on the center line and the items are located at the sides of the aisle. Hence, the recursive formula doesn't have to incorporate protections against singularities. A possible starting point for the vehicle stop location is the optimal solution to the one-dimensional rectilinear location problem with respect to the items in the pick set P_j .

Property 4.

Given the item to be picked from a stop, the optimal location of the vehicle stop assuming rectilinear travel distance is determined as follows. If the number of items in the set P_j is odd, then the location of the middle item is the optimal location. If the number of items in P_j is even, any location between the two middle items is an alternative optimal location.

Proof This is equivalent to a single facility rectilinear location problem, Francis and White (1974), Chapter 4. Q.E.D.

If the number of items in P_j is even, then the algorithm selects the midpoint between

the two middle items.

From Property 2, the order picking problem exhibits a block structure, i.e. the picks for a single stop are assigned consecutively. This reduces the number of possible combinations of item assignments to vehicle stops from $N2^N$ to $N(N+1)/2$. This makes it possible to formulate the problem as a set covering problem (SCP) with the special "consecutive-ones" structure. For more information see Segal (1974) and Bartholdi and Ratliff (1978). The model for the example of Figure 3 is given in Figure 4. Each column in the problem corresponds to a possible consecutive set of picking points assigned to a vehicle stop. The columns exhibit the "consecutive-ones" property, (i.e. the non-zero elements in each column form an uninterrupted block). If $x_k = 1$ then the stop corresponding to column k is selected. If $x_k = 0$ then the stop corresponding to column k is not chosen.

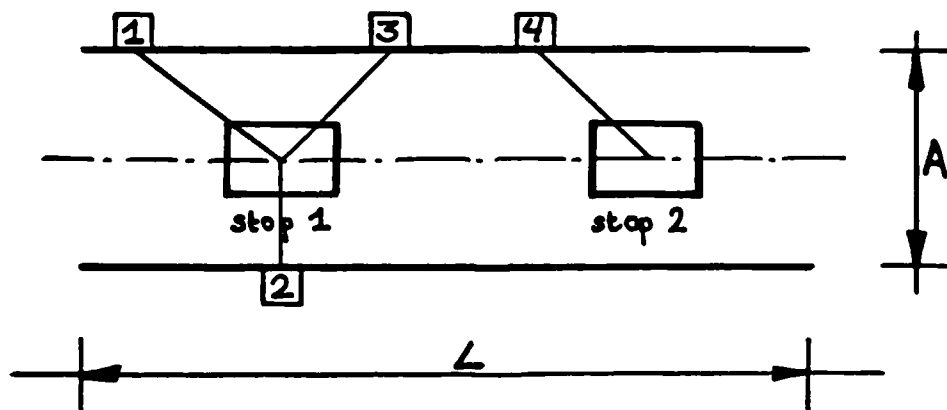


Figure 3. Wide Aisle Picking Example.

$$\text{Maximize } \sum_{k=1}^{10} c_k x_k$$

| col | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | RHS |
|-----|---|---|---|---|---|---|---|---|---|----|--|----------|
| row | | | | | | | | | | | | |
| 1 | 1 | 1 | 1 | 1 | | | | | | | $\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{10} \end{bmatrix}$ | ≥ 1 |
| 2 | | 1 | 1 | 1 | 1 | 1 | 1 | | | | | ≥ 1 |
| 3 | | | 1 | 1 | | 1 | 1 | 1 | 1 | | | ≥ 1 |
| 4 | | | | 1 | | | 1 | | 1 | 1 | | ≥ 1 |

$x_k = 0 \text{ or } 1.$

Figure 4. SCP Constraint Matrix for the Example in Figure 3.

Because of the consecutive-ones property, this set covering problem can be solved efficiently by finding the shortest path in the following acyclic graph. The items are assumed to be sorted by increasing coordinates. There exist a node in the graph for each item to be picked plus a source node (equivalent to item zero). There exists an arc in the graph corresponding to each block of consecutive picks with as length c_k . If the first item to be picked in the block is m and the last item is n , then the arc goes from node $m-1$ to node n . The shortest path from source node zero to node N gives the optimal stops and pick assignments. The graph for the example of Figure 3 is shown in Figure 5.

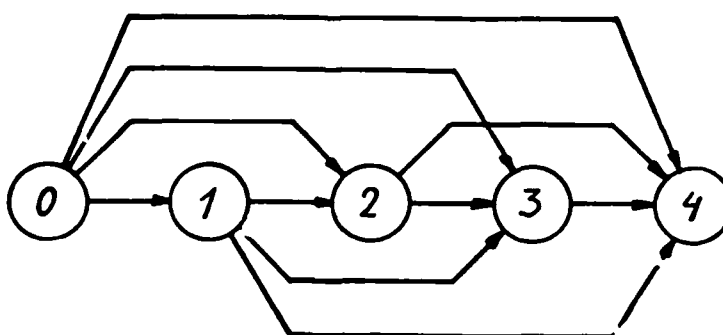


Figure 5. Graph for the Example in Figure 3.

The same algorithm can be implemented using either the rectilinear travel assumption

for the picker or the Euclidean travel assumption. The only difference is in determining the cost associated with each column in the SCP. This is determined by solving the appropriate location problem (either Euclidean or rectilinear).

This algorithm was implemented directly by dynamic programming. The number of cost computations is $O(N^2)$. If the number of iterations of the recursive formula is limited by a constant, then each cost computation requires $O(N)$ steps. Hence, the total algorithm for the Euclidean travel assumption is $O(N^3)$. The total algorithm for the rectilinear travel assumption is $O(N^2)$.

The number and location of the vehicle stops based on the Euclidean and the Rectilinear travel assumptions were compared. The number of stops was the same in 95 % of the cases and never differed by more than one stop. The picking times were compared in the following way. Under the rectilinear distance travel assumption the stops were determined based on rectilinear distance, but the picking time was computed using Euclidean distance. The two algorithms are denoted by "EUCLID" and "RECTILIN". The number of items in the aisle was set equal to 5, 10, 20 and 40. A complete description of the experimental comparison is given in Section 4. The average run time on an IBM AT microcomputer was 4.21 seconds for the EUCLID algorithm and 1.72 for the RECTILIN algorithm. The start/stop time of the vehicle didn't have a significant effect on the run times. The complete run time comparisons are shown in Table 1.

TABLE 1 Average Run Time (s)

| # picks | 5 | 10 | 20 | 40 |
|----------|------|------|------|-------|
| EUCLID | 0.04 | 0.30 | 2.00 | 14.48 |
| RECTILIN | 0.03 | 0.15 | 0.84 | 5.87 |

Note that the RECTILIN algorithm has the stops determined based on rectilinear location but the distances compared are all Euclidean.

TABLE 2 Average Percentage Increase in Travel Time Using Rectilinear Location rather than Euclidean Location

| # picks | 5 | 10 | 20 | 40 |
|---------|-----|-----|-----|-----|
| | .05 | .11 | .18 | .21 |

The small increase in travel time (less than one fourth of a percent) indicates that the actual travel assumption is not crucial. In other words, the algorithm will determine near optimal stops for any picker travel pattern which falls between the rectilinear and the Euclidean travel path. An example of such path is shown with the dotted line in Figure 6, where the Euclidean path is indicated with the dashed line and the rectilinear path with the solid line.

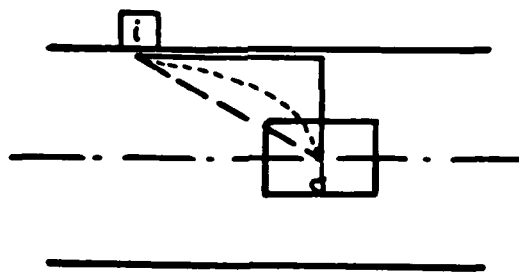


Figure 6. Various Travel Paths.

The computer run times for the RECTILIN algorithm are smaller than for the EUCLID algorithm. Hence the RECTILIN algorithm is more suitable for actual implementation in the warehouse.

3. FIXED LOCATION FOR VEHICLE STOPS

In industry, the stops are frequently not recomputed for each order, but are fixed in advanced. In other words, the vehicle is stopped in the middle of fixed block of slots if there are any picks in the block. The size of this block is called the "pattern length". The advantage of such a procedure is its simplicity and constant behavior. On the other hand, this fixed policy cannot be optimal for all orders.

It is possible to compute the optimal pattern length by continuous approximation for a given "order size". The order size is the number of locations that have to be visited in picking this order. Assume the aisle is divided into M equal blocks, each with a pattern length of x locations. Each block corresponds to a vehicle stop. Let K denote the number of locations on one side of the aisle. Then

$$x = K/M.$$

Furthermore, assume that the picking points are equally spaced on either side of the block. The optimal location of the vehicle stop is then in the middle of the block. Let L denote the aisle length, let A denote the aisle width and let u denote the width along the aisle of one location. The following relationships then exist

$$L = K \cdot u = M \cdot x \cdot u.$$

The average distance of an item location to the vehicle is then equal to

$$d = (A/4) [\sqrt{r^2+1} + (1/r) \ln(r+\sqrt{r^2+1})],$$

with the dimensionless factor r equal to

$$r = L/(MA).$$

The total time for picking the aisle is then

$$T = 2Nd/v + Mf.$$

T is a convex function of the pattern length x , as proved in Appendix A. The optimal

x^* is equal to the first x which is an integer factor of K and for which the forward difference of T becomes positive. This x^* can best be found by enumerating over increasing integer factors of K . The above procedure will be denoted by **PATTERN** and is compared with the other procedures in Section 4.

A empirical fixed pattern length of 4 slots has been observed by the authors in several distribution centers. This pattern is called the "QUAD" picking pattern by its users. This QUAD picking pattern is compared with the optimal variable stop procedures and the optimal fixed stop procedure in Section 4.

4. EXPERIMENTAL COMPARISON

Algorithms have two important characteristics. The first one is their effectiveness, this is a measure of how close their solution is to the optimal solution. The second characteristic is their efficiency, which is a measure of how many computational resources they require.

The optimal procedures using variable stops (with Euclidean and Rectilinear travel) were compared with the optimal fixed location procedure and with the empirical 4-slot fixed location procedure. A full factorial design of experiments was used. The first factor is equal to $L/(AN)$, the second factor is equal to fv/A . Both factors were chosen to be dimensionless to reduce the number of factors.

The following values were used. The length of the aisle L was 240 feet, the aisle width A was 12 feet, the walking speed v was 240 feet per minute. The slot width u was set to 4 feet, which corresponds to 60 slots on one side of the aisle. The number of items was set equal to 5, 10, 20 and 40. The stop/start time of the vehicle was set equal to 1, 0.5, 0.25 and 0.125 minutes. This corresponds to values for factor one of 4, 2, 1 and 0.5 and for factor two of 20, 10, 5 and 2.5. Hence, there exist 16 cases for each of the four procedures and for each case ten replications were run for a total of 160 experiments per algorithm. The total number of experiments was 640.

The procedures were implemented in Pascal (IBM Version 2.0 with all debugging options disabled and floating point operations interrupts) on an IBM AT with 80287 Numerical Processor. The times are for the computations of the stops only, exclude all input/output times and are given in seconds.

The PATTERN and QUAD procedures required average run times less than 0.04 seconds.

The average increase in travel time for each of the procedures in this experiment over

the time required for stop determined using the EUCLID algorithm are summarized in Table 3.

The average number of stops using each procedure is summarized in Table 4.

TABLE 3 Average Increase in Travel Time (%)

| # picks | 5 | 10 | 20 | 40 |
|-----------|-------|-------|-------|-------|
| RECTILIN | .05 | .11 | .18 | .21 |
| PATTERN | 35 | 19 | 13 | 9 |
| QUAD | 37 | 51 | 60 | 45 |
| stop time | 0.125 | 0.250 | 0.500 | 1.000 |
| RECTILIN | .24 | .15 | .10 | .06 |
| PATTERN | 25 | 22 | 13 | 15 |
| QUAD | 16 | 27 | 56 | 93 |

TABLE 4 Average Number of Stops

| # picks | 5 | 10 | 20 | 40 |
|-----------|-------|-------|-------|-------|
| EUCLID | 3 | 4 | 5 | 7 |
| RECTILIN | 3 | 4 | 5 | 7 |
| PATTERN | 2 | 3 | 6 | 7 |
| QUAD | 4 | 7 | 12 | 15 |
| stop time | 0.125 | 0.250 | 0.500 | 1.000 |
| EUCLID | 7 | 5 | 4 | 3 |
| RECTILIN | 7 | 5 | 4 | 3 |
| PATTERN | 7 | 6 | 4 | 3 |
| QUAD | 9 | 10 | 10 | 10 |

5. CONCLUSIONS

The EUCLID and the RECTILIN variable stop procedures are clearly superior in minimizing the overall travel time for picking the order. The EUCLID procedure requires a substantial amount of computation time, but the computational burden is not prohibitive. The RECTILIN procedure derives virtually the same stop locations, but its computational burden is small.

The fixed stop procedures have negligible computation times but at the cost of substantial increase in picking time. The PATTERN procedure performs better with increasing stop/start time, the QUAD procedure performs worse with increasing stop/start time. This is essentially caused by the fact that the QUAD procedure makes too many stops, which is penalized more if the stop/start time increases. The PATTERN heuristic tracks much more closely the number of stops of the optimal variable stops procedures. Its main drawback is the fact that it is only appropriate for a particular order size. In order for it to be of practical use, the order sizes of the different orders must be relative constant.

To summarize, if the vehicle stops are controlled by the computer, then either of the optimal variable stops procedure may be used effectively. However, if the computational resources are limited, then use the RECTILIN procedure. For manual systems compute the average order size of the orders and use the corresponding optimal fixed stop pattern heuristic.

6. REFERENCES

- [1] Bartholdi J. J. and H. D. Ratliff (1978). Unnetworks, with Applications to Idle Time Scheduling. Management Science, Vol. 24, 850-858.
- [2] Francis R. L. and J. A. White (1974). Facility Layout and Location. Prentice-Hall, Englewood Cliffs, New Jersey.
- [3] Goetschalckx M. and H. D. Ratliff (1985). Order Picking in an Aisle. Production and Distribution Research Report, Georgia Institute of Technology.
- [4] Segal M. (1974). The Operator-Scheduling Problem: A Network-Flow Approach. Operations Research, Vol. 22, 808-823.

7. APPENDIX A

In order to determine efficiently the optimal pattern length x^* it must be proven that the travel time T is a convex function of x . Using the notations of the paper and with the aisle width w expressed in locations equal to

$$w = A/u$$

then

$$T(x) = \frac{Nu}{2} \left[\sqrt{x^2 + w^2} + \frac{w^2}{x} \ln\left(\frac{x}{w} + \frac{1}{w} \sqrt{x^2 + w^2}\right) \right] + \frac{Kf}{x}$$

The second partial derivative of $T(x)$ is equal to

$$\frac{\partial^2 T(x)}{\partial x^2} = \frac{uNw^2}{x^3} \left[\ln(x + \sqrt{x^2 + w^2}) - \ln w + \frac{2Kf}{Nw^2 u} - \frac{x}{\sqrt{x^2 + w^2}} \right]$$

The first factor is always positive for $x \in [1, K]$, hence in order to prove convexity it must be shown that the second factor is always positive for $x \in [1, K]$. Let the second factor be denoted by $g(x)$, then

$$\frac{\partial g(x)}{\partial x} = \frac{x^2}{(x^2 + w^2)\sqrt{x^2 + w^2}} > 0.$$

The first derivative for $g(x)$ is strictly positive for $x > 0$, hence $g(x)$ is a strictly increasing function of x for $x > 0$. Also

$$\lim_{x \rightarrow 0, x > 0} g(x) = 2Kf/uNw^2 > 0.$$

Since $g(x)$ is positive for $x = 0$ and increasing, $g(x)$ is positive for $x \in [1, K]$. Hence the second derivative of $T(x)$ is positive and $T(x)$ is a convex function of x .

END

1-87

DTIC